Chapter 32: The UPD Configuration File

UPD can be configured and customized on your system using the file `updconfig`, described in this chapter. This file is usually maintained in the location `$PRODUCTS/.updfiles/updconfig`. (Its location is commonly referred to as `$UPD_USERCODE_DIR/updconfig`, where $UPD_USERCODE_DIR gets defined in the `dbconfig` file of the **UPS** database in which **UPD** is declared.) By providing default values for several variables (mostly product file and directory locations), the `updconfig` file controls where **UPD** installs products and miscellaneous product-related files.[1] It can also be used to define supplementary actions for **UPD** to perform when installing or updating products. Use of the `updconfig` file greatly reduces the amount of information the installer/maintainer needs to provide to the system for each **UPD** operation. A template `updconfig` file is available in `$UPD_DIR/ups/updconfig.template`.

Locations defined in a `updconfig` file may be overridden by specifying corresponding options on the **UPD** command line.

32.1 updconfig File Organization

The `updconfig` file always has as its first line:

```
File = updconfig
```

to identify itself to the system. The remainder of the file consists of one or more stanzas. Each stanza:

- identifies certain product instances, products or groups of products
- specifies a database on the local system in which to declare a matched product
- specifies locations on the local system in which **UPD** is to put a matched product and its related files

1. In **UPS/UPD** versions prior to and including v4_4a, the file `${UPD_USERCODE_DIR}/updusr.pm` could be used to override the default behavior of **UPD**. This file is now obsolete and can be removed.

• (optionally) lists actions for **UPS/UPD** to perform either just before or just after declaring the matched product

A `updconfig` file stanza is of the form:

```
GROUP:
...
COMMON:
...
END:
```

The `GROUP:` section of the stanza contains the product matching information. The `COMMON:` section contains the locations and actions for the matched product and its associated files. `END:` is used to end the stanza.

32.2 Product Instance Identification and Matching

There are several identifiers which can be used to specify a product instance match. When multiple values are listed for an identifier, the logical "or" of those values is used. A value can contain ${VARIABLE} strings which are expanded from the environment. Identifiers which are omitted default to `ANY`, which means they match any product instance. The following identifiers are supported:

Table 32.2.0-a:

`product`	product name
`flavor`	flavor string
`qualifiers`	qualifier string
`options`	option (anything specified via the **`-O`** (uppercase **`-o`**) option in the **UPD** command
`dist_database`	database path on the distribution server
`dist_node`	node name of distribution server

As an example, the following stanza identifies the product **exmh**, for either the flavor SunOS+5.5 or IRIX+6.3 and any qualifiers (omitted, therefore set to `ANY` by default) on *fnkits.fnal.gov*:

```
GROUP:
  product       = exmh
```

```
  flavor        = SunOS+5.5, IRIX+6.3
  dist_node     = fnkits.fnal.gov
COMMON:
  ...
END:
```

Here are two examples of using ${VARIABLE} strings:

Subscribe to changes in your local databases (e.g., if you want to do something special when, say, products are copied from one local database to another):

```
  dist_node     = file://localhost${PRODUCTS}
```

Send mail to whoever runs the upp command:

```
  mail_address      = ${USER}@fnal.gov
```

In the current implementation, the first stanza to match a given product instance is the one that gets used; **UPD** does not continue searching in the file for a "better" match.[1]

32.3 Defining Locations for Product Files

Within each stanza, file and directory locations for installing matched product instances and their associated files must be defined. These locations should be defined in terms of **UPS/UPD** read-only variables.

32.3.1 Required Locations

All the locations/keywords listed in the table below are required (the last two for distribution nodes only).

1. In the future, we hope to have a more flexible configuration file parser, more in line with **UPS** table files. We plan to make those rules upward-compatible relative to the current ones.

Note: UPS_THIS_DB, listed first, is used by **UPD** to determine the database in which to look for PROD_DIR_PREFIX (set in dbconfig, see Chapter 31: *The UPS Configuration File*). Several of the keywords that follow may be defined relative to its corresponding read-only variable ${PROD_DIR_PREFIX}.

Table 32.3.1-a:

UPS_THIS_DB	the database into which **UPS** declares the product (i.e., the directory that **UPD** specifies in the **ups declare -z** option). Recommendation: Set it to ${UPD_USERCODE_DB}, which is the database in which the updconfig file was found.
UPS_PROD_DIR	product root directory that **UPD** specifies in the **ups declare -r** option; should be defined relative to ${PROD_DIR_PREFIX} for portability
UNWIND_PROD_DIR	absolute path to directory where product gets unwound In most cases, it's ${PROD_DIR_PREFIX}/${UPS_PROD_DIR}, however in AFS and some NFS mounting configurations, products are often unwound and declared in different locations (see section 9.3 *Installing Products into AFS Space*).
UPS_UPS_DIR	ups directory that **UPD** specifies in the **ups declare -U** option, taken relative to ${UNWIND_PROD_DIR} unless an absolute path is given; usually defined as ups.
UNWIND_UPS_DIR	absolute path to directory where the ups directory gets unwound; usually defined as ${UNWIND_PROD_DIR}/${UPS_UPS_DIR} or ${UNWIND_PROD_DIR}/ups.
UPS_TABLE_DIR	table file directory that **UPD** specifies in the **ups declare -M** option **Normally this should not be set**! In some cases you may need to put the table file somewhere other than where **UPS** will automatically look (namely $PROD-UCTS/${UPS_PROD_NAME} and ${UPS_UPS_DIR}); however since UPS_TABLE_DIR must be an absolute path, the declaration becomes non-portable if you set this location.
UNWIND_TABLE_DIR	absolute path to directory where the table file gets unwound Suggestion: To maintain one table file for all flavors of a product, put it in the database; i.e., set this to ${UPS_THIS_DB}/${UPS_PROD_NAME}. To maintain each table file under $<PRODUCT>_DIR/ups, set it to ${UPS_UPS_DIR}.

Table 32.3.1-a:

UPS_TABLE_FILE	table file name that **UPD** specifies in the **ups declare -m** option Depending on where you maintain table files, choose a naming convention that identifies each file adequately. For example, if you maintain all product table files in one location, the filename should include the product name and version (e.g., ${UPS_PROD_NAME}_${UPS_PROD_VERSION}.table); if each is kept under its product root directory, the product name is not necessary (e.g., ${UPS_PROD_VERSION}.table).
UNWIND_ARCHIVE_FILE	absolute path to directory in which to unwind archive file (tar file) of product Used only on distribution server configurations.
UPS_ARCHIVE_FILE	archive file (tar file) location that **UPD** specifies in **ups declare -T ftp://host${UPS_ARCHIVE_FILE}** Used only on distribution server configurations.

32.3.2 Read-Only Variables Usable in Location Definitions

The following predefined **UPS/UPD** read-only variables can be used in the definition of locations described above. These variables get their values from the command line and/or the dependency list.

Do not try to redefine these variables. When you use these variables, always enclose them in curly brackets ({}) as shown in the list.

Table 32.3.2-a:

${UPS_USERCODE_DB}	database containing **UPD** configuration
${UPS_USERCODE_DIR}	directory containing **UPD** configuration
${UPS_PROD_NAME}	name of product
${UPS_PROD_FLAVOR}	flavor of product
${UPS_PROD_QUALIFIERS}	qualifiers of product
${UPS_BASE_FLAVOR}	flavor trimmed as in **ups flavor -1**
${DASH_PROD_FLAVOR}	flavor with non-word characters replaced by dashes; e.g., if {UPS_PROD_FLAVOR} is set to SunOS+5, then {DASH_PROD_FLAVOR} has the value SunOS-5. This is used to avoid problems with unusual symbols in file and directory names.

Table 32.3.2-a:

${DASH_PROD_QUALIFIERS}	qualifier list with non-word characters replaced by dashes; e.g., if {UPS_PROD_QUALIFIERS} is `qual1+qual2`, then {DASH_PROD_QUALIFIERS} is `qual1-qual2`. This is used to avoid problems with unusual symbols in file and directory names.
${SUFFIX}	suffix of archive file; e.g., `tar`, `zip`, etc. Used only on distribution server configurations.
${PROD_DIR_PREFIX}	`PROD_DIR_PREFIX` of database, defined in **UPS** configuration file `dbconfig` (see Chapter 31)

32.3.3 Sample Location Definitions

The following partial stanza, taken from the example in section 32.5.1, shows several location specifications:

```
COMMON:
     UPS_THIS_DB  = "${UPD_USERCODE_DB}"
                                          UPS_PROD_DIR        =
"${UPS_PROD_NAME}/${UPS_PROD_VERSION}/${DASH_PROD_FLAVOR}
                     ${DASH_PROD_QUALIFIERS}" (this must be all on one
line in the real file)
  UNWIND_PROD_DIR = "${PROD_DIR_PREFIX}/${UPS_PROD_DIR}"
      UPS_UPS_DIR = "ups"
   UNWIND_UPS_DIR = "${UNWIND_PROD_DIR}/${UPS_UPS_DIR}"

   #    Default (do not actually set UPS_TABLE_DIR):
   #   UPS_TABLE_DIR = "${UPS_THIS_DB}/${UPS_PROD_NAME}"
    UNWIND_TABLE_DIR = "${UPS_TABLE_DIR}"
      UPS_TABLE_FILE = "${UPS_PROD_VERSION}.table"
...
   END:
```

32.4 Pre- and Postdeclare Actions

An *action* is a construction that identifies a **UPS** or user-defined operation via the ACTION keyword, and lists functions to perform, in addition to any internal processes, when the operation is executed. An action stanza has the format:

```
ACTION=<VALUE>
   <function_1>([<argument_1>] [, <argument_2>] ...)
```

```
<function_2>([<argument_1>] [, <argument_2>] ...)
...
```

The `updconfig` file uses actions to define the steps **UPD** is to perform during an installation/update of the matched product instance(s). The ACTION keyword values indicate when to perform the steps (before or after issuing the **`ups declare`** command), and the steps themselves are listed as functions under the ACTION line. In a `updconfig` file, actions can be listed anywhere in the `COMMON:` part of a stanza.

32.4.1 ACTION Keyword Values

Currently two action keyword values are supported for use in `updconfig`:

Table 32.4.1-a:

`predeclare`	Perform listed functions after product files have been unwound, but before the product has been declared
`postdeclare`	Perform listed functions after product has been declared

Functions are then listed after the ACTION keyword line, using the following syntax:

```
Action = predeclare
    function(arg,arg,...)
    function(arg,arg,...)
```

32.4.2 The execute Function

Currently, only the **`execute`** function is supported for use in `updconfig`:

`execute("<command>", <UPS_ENV_FLAG>, [, <VARIABLE>])`

It executes a shell-independent command and (optionally) assigns the output to an environment variable, **`<VARIABLE>`**. It takes a required parameter (**`UPS_ENV_FLAG`**) which indicates whether to define **UPS** local variables. This parameter can take the following values:

UPS_ENV	define all local **UPS** environment variables before sourcing (the script or command relies on these being defined)
NO_UPS_ENV	do not define the local **UPS** environment variables (the script or command doesn't use them)

If the optional third argument, **`<VARIABLE>`**, is not specified, then the specified command is executed but the output from that command is not saved. This command does not have to be shell-independent.

For example, say that you want to make group-writable the directory in which a product has been unwound before it is declared. You would include the action:

```
ACTION = PREDECLARE
    execute ("chmod -R g+w ${UNWIND_PROD_DIR}", NO_UPS_ENV )
```

32.5 Examples

32.5.1 Generic Template updconfig File

This example is taken from the template, `$UPD_DIR/ups/updconfig.template` (comments are included in the actual template file). The template is designed to be usable *as is*, if:

- you have only one **UPS** database
- you want your product root hierarchy to be:

  `${PROD_DIR_PREFIX}/<product>/<version>/<flavor><qualifiers>`
- you want your table files to reside in the **UPS** database as:

  `${UPS_THIS_DB}/<product>/<version>.table`

If your requirements are different, this file is still useful as a starting point from which to make modifications.

```
File = updconfig
#
GROUP:
  product       = ANY
  flavor        = ANY
  qualifiers    = ANY
  options       = ANY
  dist_database = ANY
  dist_node     = ANY

COMMON:
     UPS_THIS_DB  = "${UPD_USERCODE_DB}"
                                          UPS_PROD_DIR        =
"${UPS_PROD_NAME}/${UPS_PROD_VERSION}/${DASH_PROD_FLAVOR}
                     ${DASH_PROD_QUALIFIERS}" (no line break in real
file)
  UNWIND_PROD_DIR = "${PROD_DIR_PREFIX}/${UPS_PROD_DIR}"
      UPS_UPS_DIR = "ups"
   UNWIND_UPS_DIR = "${UNWIND_PROD_DIR}/${UPS_UPS_DIR}"

#    Default (do not actually set UPS_TABLE_DIR):
#   UPS_TABLE_DIR = "${UPS_THIS_DB}/${UPS_PROD_NAME}"
 UNWIND_TABLE_DIR = "${UPS_TABLE_DIR}"
   UPS_TABLE_FILE = "${UPS_PROD_VERSION}.table"
#
#    Possible alternative, where the table files live
#    within the product's ups directory.  Note,
#    in this case you ALSO should not set UPS_TABLE_DIR.
```

```
#
##  UPS_TABLE_DIR = "${UNWIND_UPS_DIR}"
#UNWIND_TABLE_DIR = "${UPS_TABLE_DIR}"
# UPS_TABLE_FILE  = "${UPS_PROD_NAME}.table"
#
#  ACTION = PREDECLARE
#           add functions
#  ACTION = POSTDECLARE
#           add functions
END:
```

32.5.2 Distribution from the fnkits Node Only

As a second example, we show the `GROUP:` portion of a file that specifies a particular distribution host. Aside from the `dist_node` entry, the stanza is identical to that of the template `updconfig` file, and therefore applies to any products coming from the specified host, *fnkits.fnal.gov. fnkits* is the central Computing Division product server, and there are several names for it. All the names are all listed and delimited by colons.

```
File = updconfig

GROUP:

  product       = ANY
  flavor        = ANY
  qualifiers    = ANY
  options       = ANY
  dist_database = ANY
              dist_node                                          =
fnkits:fnkits.fnal.gov:kits:kits.fnal.gov:upd:upd.fnal.gov

COMMON:
...
END:
```

This **UPD** configuration file could be expanded to include additional stanzas to accommodate products from other distribution nodes.

32.5.3 Customized Treatment of ups Directory and Table Files

In this example, the distribution node again has changed relative to the template, and this time there are also changes in the `COMMON:` section. The distribution node is `e007.dist.xyz.edu`. UPS_PROD_DIR is no longer defined relative to PROD_DIR_PREFIX, but is now placed under the

`/e007/base_code` directory. All the table files are placed in a single directory (`/e007/table_files`), therefore the table file names must include the product name in order to be identifiable. Here they will be named `<product>_<version>.table` (defined using the corresponding variables as `${UPS_PROD_NAME}_${UPS_PROD_VERSION}.table`).

```
File = updconfig

GROUP:
  product       = ANY
  flavor        = ANY
  qualifiers    = ANY
  options       = ANY
  dist_database = ANY
  dist_node     = e007_dist.xyz.edu

COMMON:
      UPS_THIS_DB      = "${UPD_USERCODE_DB}"
                          UPS_PROD_DIR                          =
"/e007/base_code/${UPS_PROD_NAME}/${UPS_PROD_VERSION}/

${UPS_PROD_FLAVOR}${UPS_PROD_QUALIFIERS}" (no line break in real file)
     UNWIND_PROD_DIR  = "${PROD_DIR_PREFIX}/${UPS_PROD_DIR}"
     UPS_UPS_DIR      = "ups"
     UNWIND_UPS_DIR   = "${UNWIND_PROD_DIR}/${UPS_UPS_DIR}"
     UPS_TABLE_DIR    = "/e007/table_files"
     UNWIND_TABLE_DIR = "${UPS_TABLE_DIR}"
                             UPS_TABLE_FILE                     =
"${UPS_PROD_NAME}_${UPS_PROD_VERSION}.table"
END:
```

32.5.4 Implementing Multiple Configurations

Here is an example that shows how to configure the file if more than one database and distribution node are used. The first section instructs **UPD** where to unwind products that are distributed from *fnkits* and how to declare them. The second section, with different naming conventions and file hierarchies, instructs **UPD** where to unwind and how to declare products obtained from the CDF distribution node *cdf-kits.fnal.gov.*

```
File = updconfig

GROUP:

  product       = ANY
  flavor        = ANY
  qualifiers    = ANY
```

```
  options       = ANY
  dist_database = ANY
              dist_node                                    =
fnkits:fnkits.fnal.gov:kits:kits.fnal.gov:upd:upd.fnal.gov

COMMON:

     UPS_THIS_DB      = "${UPD_USERCODE_DB}"
                        UPS_PROD_DIR                       =
"${UPS_PROD_NAME}/${UPS_PROD_VERSION}/${UPS_PROD_FLAVOR}
                           ${UPS_PROD_QUALIFIERS}"      (no line
break in real file)
     UNWIND_PROD_DIR  = "${PROD_DIR_PREFIX}/${UPS_PROD_DIR}"
     UPS_UPS_DIR      = "ups"
     UNWIND_UPS_DIR   = "${UNWIND_PROD_DIR}/${UPS_UPS_DIR}"
###  UPS_TABLE_DIR    = "${UPS_THIS_DB}/${UPS_PROD_NAME}"
     UNWIND_TABLE_DIR = "${UPS_TABLE_DIR}"
     UPS_TABLE_FILE   = "${UPS_PROD_VERSION}.table"

END:

GROUP:

  product       = ANY
  flavor        = ANY
  qualifiers    = ANY
  options       = ANY
  dist_database = ANY
  dist_node     = cdf-kits.fnal.gov

COMMON:

     UPS_THIS_DB      = "~cdfsoft/declare"
                        UPS_PROD_DIR                       =
"${UPS_PROD_NAME}/${UPS_PROD_VERSION}/${UPS_PROD_FLAVOR}
                          ${UPS_PROD_QUALIFIERS}"   (no line break
in real file)
                                UNWIND_PROD_DIR            =
"/cdf/products/${UPS_PROD_NAME}/${UPS_PROD_VERSION}"
     UPS_UPS_DIR      = "ups"
     UNWIND_UPS_DIR   = "${UNWIND_PROD_DIR}/${UPS_UPS_DIR}"
###  UPS_TABLE_DIR    = "${UPS_THIS_DB}/${UPS_PROD_NAME}"
     UNWIND_TABLE_DIR = "${UPS_TABLE_DIR}"
     UPS_TABLE_FILE   = "${UPS_PROD_VERSION}.table"

END:
```

32.5.5 Sample Configuration for AFS Space Using ACTIONS

In AFS space, you may need to release the read-write volume before you can declare a product, as discussed in section 9.3 *Installing Products into AFS Space*. For this you would use a PREDECLARE action. You may also need to release the read-write **UPS** database after the product is declared, which can be done in a POSTDECLARE action. These actions are shown in this example.

```
File = updconfig

 GROUP:
  product       = ANY
  flavor        = ANY
  qualifiers    = ANY
  options       = ANY
  dist_database = ANY
  dist_node     = ANY

COMMON:
     UPS_THIS_DB = "/afs/.fnal.gov/ups/db"
                                          UPS_PROD_DIR        =
"${UPS_PROD_NAME}/${UPS_PROD_VERSION}/${UPS_PROD_FLAVOR}
                    ${UPS_PROD_QUALIFIERS}"   (no line break in real
file)
  UNWIND_PROD_DIR = "/afs/.fnal.gov/ups/${UPS_PROD_DIR}"
      UPS_UPS_DIR = "ups"
   UNWIND_UPS_DIR = "${UNWIND_PROD_DIR}/${UPS_UPS_DIR}"
 UNWIND_TABLE_DIR = "${UPS_THIS_DB}/${UPS_PROD_NAME}"
   UPS_TABLE_FILE = "${UPS_PROD_VERSION}.table"

  ACTION = PREDECLARE
                      Execute("/usr/local/bin/upd_volrelease
${UNWIND_PROD_DIR}", NO_UPS_ENV)
  ACTION = POSTDECLARE
     Execute("/usr/local/bin/upd_volrelease ${UPS_THIS_DB}",
NO_UPS_ENV)

END:
```

32.5.6 Distribution Node Configuration

For this example, we present an abridged version of the `updconfig` file used on *fnkits*. The real one is fairly long and repetitive (will be shortened mid-2000), and is described in Chapter 22: *Configuration of the fnkits Product Distribution Node*

fnkits has a local database containing products destined for use on that node in addition to the `KITS` distribution database. On *fnkits*, one `updconfig` is used for both databases (this too will change mid-2000). It resides under the local database, and ${UPD_USERCODE_DIR} is defined accordingly in the `dbconfig` files for both databases.

The `updconfig` file on *fnkits* includes several stanzas, each of which pertains to a category of product. The product-matching criterion for each stanza is an `options=<option>` line which indicates the category[1]. This example shows only the stanzas used for ordinary distributed products (the default) and locally installed products. The default stanza is identified by the absence of an option, and the local stanza is identified by the option `local`.

The default stanza includes a PREDECLARE and a POSTDECLARE action. The PREDECLARE action contains a set of **`execute`** statements to **`chmod/chgrp`** the files to the right group id and permissions, and another set to symlink files under `/ftp/KITS` to provide the old-style (**UPS/UPD** v3) `KITS` hierarchy[2] of `KITS/Flavor/product/version`. The POSTDECLARE action makes a convenience tar file of the `ups` directory for users downloading via **FTP**. The stanza for `local` products contains no actions.

- Many of the location definitions and functions are quite long, and are shown here on multiple lines for readability.
- In the real file, each definition or function must be contained on a single line.

```
File=updconfig
#
group:
    # normal, ordinary products added to kits
   common:
    # actual locations of things
    UPS_THIS_DB = "/ftp/upsdb"

UNWIND_PROD_DIR="/ftp/products/${UPS_PROD_NAME}/${UPS_PROD_V
ERSION}/

${UPS_PROD_FLAVOR}/${UPS_PROD_NAME}_${UPS_PROD_VERSION}_
      ${UPS_PROD_FLAVOR}${UPS_PROD_QUALIFIERS}"
    UNWIND_UPS_DIR = "${UNWIND_PROD_DIR}/ups"
```

---

1. For this type of configuration, unless some automatic implementation of option-matching is implemented (as is the case on *fnkits*), a product provider would need to include the appropriate option as **`upd addproduct -O <option>`** when adding the product to the distribution node, in order to invoke the right stanza. The option `local` is an exception: the person installing a product for local use on the distribution node would need to use **`upd install`** with the **`-O local`** option.
2. Currently nothing prunes old links or files from this hierarchy.

```
                                    UNWIND_TABLE_DIR        =
"/ftp/products/${UPS_PROD_NAME}/${UPS_PROD_VERSION}/
      ${UPS_PROD_FLAVOR}"
    UNWIND_ARCHIVE_FILE = "${UNWIND_PROD_DIR}.${SUFFIX}"
    #
    # declared values of things
                                     UPS_TABLE_FILE         =
"${UPS_PROD_NAME}_${UPS_PROD_VERSION}_${UPS_PROD_FLAVOR}
       ${UPS_PROD_QUALIFIERS}.table"
    UPS_TABLE_DIR = "${UNWIND_TABLE_DIR}"
    UPS_PROD_DIR = "${UNWIND_PROD_DIR}"
    UPS_UPS_DIR = "ups"
    UPS_ARCHIVE_FILE = "${UNWIND_ARCHIVE_FILE}"

    action = predeclare
        #
        # fix group permissions
                Execute("chgrp  upd  ${UNWIND_TABLE_DIR}/*",
NO_UPS_ENV)
               Execute("chmod  o-rwx  ${UNWIND_TABLE_DIR}/*",
NO_UPS_ENV)
            Execute("chmod  a+r  ${UNWIND_TABLE_DIR}/*.table",
NO_UPS_ENV)
       #
       # make old-KITS compatable hierarchy files

                                          Execute("test     -d
/ftp/KITS/${UPS_BASE_FLAVOR}/${UPS_PROD_NAME}/
                             ${UPS_PROD_VERSION}  ||  mkdir  -p
/ftp/KITS/${UPS_BASE_FLAVOR}/
                        ${UPS_PROD_NAME}/${UPS_PROD_VERSION}",
NO_UPS_ENV)

                                                  Execute("cd
/ftp/KITS/${UPS_BASE_FLAVOR}/${UPS_PROD_NAME}/${UPS_PROD_VER
SION};
          rm -f ${UPS_PROD_NAME}_${UPS_PROD_VERSION}_
                ${UPS_PROD_FLAVOR}${UPS_PROD_QUALIFIERS}.*",
NO_UPS_ENV)

                                                  Execute("cd
/ftp/KITS/${UPS_BASE_FLAVOR}/${UPS_PROD_NAME}/${UPS_PROD_VER
SION};
                 /usr/bin/ln  -fs  ${UNWIND_PROD_DIR}.*  .  ||
true",NO_UPS_ENV)

    action = postdeclare
#
```

```
# Make a xxx.ups.tar file
            Execute("test  -d  \"${UNWIND_UPS_DIR}\"  &&  cd
${UNWIND_UPS_DIR} &&
            tar  cf  ${UNWIND_PROD_DIR}.ups.tar  .  ||  true",
NO_UPS_ENV)
end:

group:
    #
    # products installed locally
    options = "local"

common:
    #
    # actual locations of things on local system
    UPS_THIS_DB = "/fnal/ups/db"
                                      UNWIND_PROD_DIR         =
"/fnal/ups/${UPS_PROD_NAME}/${UPS_PROD_VERSION}/
      ${UPS_PROD_FLAVOR}${UPS_PROD_QUALIFIERS}"
    UNWIND_UPS_DIR = "${UNWIND_PROD_DIR}/ups"
    UNWIND_TABLE_DIR = "${UPS_THIS_DB}/${UPS_PROD_NAME}"
    UPS_TABLE_FILE = "${UPS_PROD_VERSION}.table"
    #
    # declared values of things
    UPS_PROD_DIR = "${UNWIND_PROD_DIR}"
    UPS_UPS_DIR = "ups"
end:
```